

12 **EUROPEAN PATENT APPLICATION**

21 Application number: 88300645.4

51 Int. Cl.4: **G11B 7/013 , G11B 20/12 ,
G11B 27/28**

22 Date of filing: 26.01.88

43 Date of publication of application:
02.08.89 Bulletin 89/31

64 Designated Contracting States:
DE FR GB IT NL

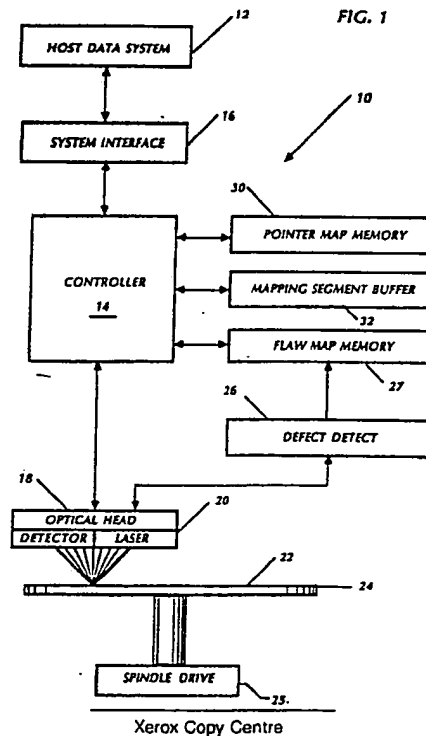
71 Applicant: **LASERDRIVE LTD.**
1101 Space Park Drive
Santa Clara California 95054(US)

72 Inventor: **Williams, Christopher Denis**
529 Encino Drive
Aptos California 95003(US)

74 Representative: **Cross, Rupert Edward Blount**
et al
BOULT, WADE & TENNANT 27 Furnival Street
London EC4A 1PQ(GB)

54 **Data storage system.**

57 A write-once, read-many optical data storage system (10) includes an optical disk (24) to which data is written such that a host data system (12) is enabled to access any one of a predetermined number of logical addresses for storing data at a selected logical address whether or not data had previously been stored at that logical address. The system allows the write-once optical disk to appear to a host system user to be a read/write disk.



DATA STORAGE SYSTEM

The present invention relates generally to the field of computer data storage systems and more specifically to the field of optical disk data storage systems.

Data storage systems are well known in the art and are commonly used in computer systems where data that is generated or manipulated must be stored and/or retrieved at will. Two major categories of data storage systems that are commonly used to store and retrieve large blocks of data include magnetic disk storage systems and optical disk storage systems. Each category has advantages and disadvantages.

Magnetic disk storage systems are usually used in applications requiring frequent erasure of stored data and replacement with updated data. An advantage of many such systems is that the magnetic storage media is removable, e.g. in floppy disk systems. Only lower capacity and slower access time versions of such systems are generally of this removable type, however. A key disadvantage of magnetic disks is that they are susceptible to mechanical damage as well as inadvertent erasures.

Optical disk systems typically are used where very large amounts of data are to be stored, e.g. 100 megabytes or more. Although optical disks generally are removable, most have the disadvantage of being read-only storage media. The optical disks are prerecorded in a conventional manner, such as using an ablative-pit method in which a precisely focused laser beam burns a depression or pit in the sensitive recording layer of the disk surface. In the optical disc reader, as the disk surface is passed under a light beam in the read mode, a photodetector senses the presence or absence of the depressions and emits electrical signals which are transformed into digital data bits.

Write-once optical disk data storage systems are now available, and offer very high data writing and reading capacity in a removable storage medium. However, such systems have the drawback that once data has been written into a sector of an optical disk, this data cannot be updated by rewriting the sector. Many attempts have been made in the art to enable the high storage capacity of write-once optical disk memories to be utilized efficiently in a computer system in a manner analogous to magnetic disk storage systems, wherein data in a given sector can be updated at will. All such attempts have had inherent problems.

A key problem relates to how the directory or index of stored data on the optical disk is maintained. Without such directory information, it is impossible to selectively access and retrieve data on the disk. In one approach, when data is stored on the optical disk its location is maintained in some sort of directory or index stored on a companion magnetic floppy or magnetic hard disk. This approach has the critical disadvantage of the susceptibility to loss of the directory by erasure or mechanical damage to the magnetic disk, which results in complete and irretrievable loss of the ability to selectively retrieve data stored on the optical disk. This is in addition to the need for keeping each magnetic directory disk physically associated with the optical disk whose directory it is storing. The loss of either of these media renders the other useless.

Another approach has been to store the optical disk directory on the optical disk and, when the disk is first inserted into the disk storage system, initializing an associated magnetic memory to correspond to this directory. As new data is written onto the disc, only the magnetic memory is updated. The entire magnetic memory version of the directory is rewritten onto the optical disk immediately prior to the removal of the disk or the powering down of the system. The disadvantage of this approach is that it uses much disk space since the entire directory may be written on the disk many times, and it is vulnerable to loss of the directory by power interruption before the directory has been rewritten onto the optical disk.

A third alternative method relies on address pointer fields associated with each data segment written to the disk. When data is written onto a particular segment of the optical disk, the associated pointer field remains blank. When an update of the data is desired, it is written to a different segment and the physical address, i.e. the actual physical location on the disk where the updated data has been written, is written to the pointer field of the original data segment. When data is to be read, the pointer field of the original data segment is read to determine the address of the updated information, if any. Because there may be a series of updates to updates in such a pointer field system, trying to read the most recent update of a desired data segment may result in a lengthy sequential examination of a trail of updated data segments before the current data segment is found. The inherent slowness of this system has the serious additional disadvantage of becoming worse and worse as more data updates are made on the optical disk.

What is needed in the art is a system whereby a map of where various data segments are written on the optical disk can be easily maintained and updated on the disk and whereby permanently written data on an optical disk can be used to construct an audit trail of all prior written data segments in the event of the loss of some portion of the disk map. Such a system would increase retrieval speed and provide, in effect,

a pseudo-erasable optical disk storage system, a system that emulates an erasable magnetic storage medium while providing the high density storage advantages and data permanence of an optical disk.

The design of optical storage systems presents another inherent problem. Because the recording surface of an optical disk may have defects and because data cannot be retrievably written to a defective surface, it is necessary for the optical system to detect such defects and avoid writing to such areas. Typically, this is accomplished by making two passes over an area on the surface of the disk during each writing operation; once to write the data and once to verify the actual writing. This slows down the operation of the system considerably.

According to this invention there is provided an optical data storage system for enabling the writing of data to and the reading of data from a write-once, read-many optical disk by a host system such that said host system is enabled to access any one of a predetermined number of logical addresses for storing host data whether or not host data has previously been stored at that logical address, characterised by a write-once optical disk including at least one recording surface divided into a plurality of sequential data storage segments, each having a host data portion and a logical address portion, and having the physical location of each of said data storage segments represented by a physical address different from the physical address of any other data storage segment; and means for writing each successive host data to the host data portion of a next sequential unwritten data storage segment, including means for writing to the logical address portion of said next sequential data storage segment the logical address specified by said host system for said host data.

The invention provides an optical disk data storage system which allows a write-once optical disk to appear to a host system user to be a read/write disk.

The system enables rapid identification and retrieval of data at the most recently written physical address on an optical disk corresponding to a selected host system logical address, and also provides means for recovering lost directory information for any optical disk.

Further, defects in an optical disk recording surface can be automatically detected and bypassed when writing to or recording data from the optical disk.

Preferably the optical disk includes a plurality of sequential mapping segments each for storing at least one logical address; and means responsive to said host data writing means for writing to the next sequential unwritten mapping segment the logical address specified by said host data system for said host data being written by said host data writing means.

Preferably the system includes means for determining the last physical address into which host data was written having said specified logical address, and means for reading the host data stored in the data storage segment at said last physical address.

Preferably the system includes a random access magnetic pointer map memory which stores a map of all available data logical addresses and their correspondence to the most recent physical addresses of the data storage segments on the associated write-once optical disk wherein host data has been written. This pointer map is generated on power up of the system and each time a new optical disk is inserted into the data storage system. All written sequential mapping segments on the disk are read as part of this initialization process.

To speed up the disk writing process and to minimize wasted disk space, a magnetic mapping segment buffer can be provided to temporarily store a record of all recent physical address locations on the optical disk written into since the last time the disk mapping segments were updated, along with the correspondence between logical address of the last data stored at each such physical address. When full of data, the buffer is flushed to transfer this record to the next sequential series of unwritten mapping segments on the optical disk.

The system can include means for detecting the location of defects in the recording surface of the optical disk and for storing the location of these defects in a predetermined segment of said optical disk. This information enables flawed areas of the disk to be bypassed when the system is writing host data on the disk. A separate magnetic flaw map memory may be included to keep track of defects arising after the original defect detection process has been performed.

The system of the invention has the advantage of emulating erasable magnetic memories while retaining all the characteristics of optical data storage systems, including their large data storage capacity and the permanence of stored data. The system of the invention also has the advantage of enabling the retrieval of disk directory information if the directory is ever lost or destroyed.

This invention will now be described by way of example with reference to the drawings, in which:-

Fig. 1 is a block diagram of the main components of an optical data storage system according to the invention;

Fig. 2 is a diagram showing the division of the recording surface of an optical disk into mapping segments, data storage segments, and a flaw map;

Fig. 3 is a flow chart showing the sequence of steps included in the initialization operation of the data storage system of Fig. 1;

5 Fig. 4 is a flow chart showing the sequence of steps included in the write operation of the data storage system of Fig. 1; and;

Fig. 5 is a flow chart showing the sequence of steps included in the read operation of the data storage system of Fig. 1 when it retrieves only the most recently written host data for a selected logical address

10

A pseudo-erasable write-once, read-many optical data storage system 10 according to the present invention is shown in block diagram form in FIG. 1. The storage system 10 is accessed by a host data system 12, such as a computer. Host system 12 either couples data to storage system 10 for storage at an address (defined as a "logical" address) specified by the host system or receives data stored at that address depending on whether a write or a read command has been generated by the host system. These
15 commands are coupled to a controller means 14 in system 10 via a system interface 16. In the preferred embodiment, system interface 16 enables the system 10 to be accessed in a standard manner by the host system 12 over a bus shared by other peripheral devices. Usually, interface 16 uses "logical" rather than "physical" addressing. Even if the host system thinks it is addressing a physical address, the system will
20 treat it as a logical address in the manner as explained below. All data is addressed by the host system 12 as logical blocks up to the maximum number of blocks available in a device. The actual physical address of any host system data in storage system 10 is not known to the host data system 12. An exemplary system interface 16 is the Small Computer Systems Interface (SCSI), known in the art.

Controller 14 controls an optical head 18 having a writing means, comprising a laser 20, for writing data
25 to the recording surface 22 of a write-once optical disk 24. The data so written is permanently recorded on the recording surface 22; that is, the data is not intended to be erasable or be overwritten. The laser 20 is directed to selected write positions on the surface 22 via conventional movement of optical head 18 under the control of controller 14 and via rotation of optical disk 24 via a spindle drive 25. The system 10 also preferably includes means 26 for detecting defects in the recording surface 22 of the optical disk 24 and for
30 retaining the location of these defects in a flaw map memory 27. Flaw map 27 stores the location of defects at a predetermined storage location on the recording surface 22. Controller 14 includes means for preventing the writing of data to an area of the recording surface 22 on which a defect has been detected as specified by flaw map 27.

The system also includes a magnetic pointer map memory means 30, which preferably is a conventional random access memory (RAM). Pointer map 30 acts as a look-up table to keep track of where host
35 data at each logical address are actually written on the optical disk. When the host data system 12 wishes to rewrite host data at a specified logical address, the new host data is actually written to the next sequential unwritten area of the optical disk, and the pointer map 30 is updated to reflect this new physical location for finding host data at the specified logical address. A key aspect of the operation of the system is
40 that host data is always written to the optical disk 24 as if the disk 24 were a magnetic tape. That is, new host data is always written sequentially to the optical disk 24 beginning at the end of the last data storage segment written on disk 24. As described in greater detail below, the logical address of a given host data block is also stored on the optical disk 24 as part of each data segment written to the optical disk 24. Preferably, these logical addresses are also stored on the optical disk in a separate area on disk 24 as
45 sequential mapping segments for enabling rapid initialization of pointer map 30 each time the new optical disk 24 is inserted into the system 10. As seen below, this greatly simplifies the initialization of pointer map 30.

More specifically, pointer map 30 contains a list of all data logical addresses available for storage of data by the host data system 12. Pointer map 30 further stores a map of the physical address of the data
50 storage segment on the associated optical disk 24 wherein the most recent update of host data has been written for each logical address. For any logical address wherein the host data system 12 has not yet recorded data, no physical address is yet stored in the pointer map 30. An entry of 0 in pointer map 30 at any logical address indicates that data having this logical address has not yet been written on the optical disk. Controller 14 on power-up of the system 10, and each time a new optical disk 24 is inserted into the
55 system 10, first reads all written sequential mapping segments on the disk 24 for reinitializing pointer map 30. The organization of these mapping segments and the way this initialization is performed is described in greater detail hereinbelow.

The optical disk is organized into a plurality of segments or sectors each having the ability to store a

predetermined number of bytes of data, e.g. 512 bytes. Each sector is usually only available to be written to as part of a single write sequence. Since it would be inefficient to store a logical address in a separate location on the disk each time a host data block is written to the disk, it is preferable that the history of the most recent write transactions on the surface of the disk be temporarily stored in a random access memory in system 10. Mapping segment buffer 32 provides this memory storage means. Mapping segment buffer 32 keeps track of the sequential physical addresses in which host data has been written in the data segments corresponding logical address for each data segment stored. Preferably, mapping segment buffer 32 is full when a sector's worth of data is contained therein. At that point, the buffer 32 is flushed by controller 14 to transfer this data to the next sequential unwritten mapping segment on the optical disk 24. Note that it is important that this mapping data be written to the same optical disk as the host data is written.

Should the buffer 32 not be filled when power is lost or when the user removes the optical disk 24 from the system 10, the mapping data is not lost nor is the ability lost to access selectively the most recent host data stored on the disk 24. This is because, as mentioned above, a separate audit trail has been written onto optical disk 24 during writing of each host data to a given data storage segment. When a host data block is stored in a data storage segment, a part of the storage segment is reserved for storage of the logical address of this host data. Thus, if the data in the magnetic segment buffer 32 is lost, controller 14 can read the sequentially written data storage segments that have been written on disk 24 since the last buffer 32 contents were written to the disk, to regenerate the mapping data lost.

Referring to FIG. 2, shown is an exemplary recording surface 22 of an optical disk 24, including 30 segments for the writing of data. In practice, as described below, a given optical disk may include hundreds of thousands of segments for the storage of data. As seen, the recording surface 22 shown in FIG. 2 is divided into mapping segments or sections 34 and data storage segments or sections 36, and one or more data storage segments 38 arranged in a spiral manner. Thus, for example, segments 1 and 2 would store disk flaw data, segments 3-10 would be available for storage of mapping data 34 and the remaining segments 11-30 would be available as data storage segments 36. With this segment organization, all segments have the same length so that the longer outer tracks of the recording surface 22 contain more segments than do the tracks nearer the center of the disk 24. This constant linear velocity (CLV) organization means that the relative speed with which data is read from the disk 24 remains the same as the reading progresses from the outer edge of the disk 24 inward. This arrangement allows data to be packed on outer tracks just as tightly as on inner tracks, thereby greatly increasing the storage capacity of system 10.

Alternative arrangements of mapping segments 34 and data storage segments 36 on disk 24 are possible. For example, constant angular velocity (CAV) can be achieved by making segments on outer tracks physically larger than those on inner tracks. This results in a disk organized as concentric rings rather than as a continuous spiral of segments.

Another alternative arrangement would be where the mapping segments are positioned into several separate areas on the disk surface 22. This arrangement enables faster writing of host data where the mapping segment data is not initially buffered in a buffer such as buffer 32, since a given mapping segment is nearer to the segment into which data is being written, but it requires additional seeking of mapping segments during initialization of the pointer map 30. Although one could go still farther, and rely just on recording the logical address after each host data is written to a given data storage segment, eliminating the separate mapping segments entirely, this approach adds two disadvantages which would be significant in most application. First, every single sector or segment would then need to be read out to complete initialization of the pointer map 30. This initialization process might take as long as 45 minutes for a standard sized optical disk. The second disadvantage is that this approach eliminates the ability of the system 10 to be resistant to loss of mapping data written on the disk. According to the preferred embodiment, with the logical addresses stored both with each respective host data block and in the separate mapping segments, the loss of a given mapping segment and the data therein, due to disk damage, for example, would be recoverable by reading the logical addresses stored in the corresponding data segments.

As explained above, the system requires that some disk space be reserved for use as a sequential history of every write transaction that has taken place on the surface of the disk 24. This disk space comprises mapping segments 34. The amount of space needed for this write transaction history is dependent upon the number of sectors on the disk surface which can be written. As mentioned below, one or more techniques will be available to lessen the amount of sector space required for the storage of this history. Preferably, the write transaction history is kept on the optical disk 24 in one predetermined contiguous area preceding the host data system user data space containing a plurality of mapping segments 34.

The information stored in mapping segments 34 comprises a sequential table of each logical address of a host data block written to the disk 24. Since the physical address on disk 24 of the first data storage segment 36 is known, and all subsequent data storage segments are preferably written sequentially after the first data storage segment on the optical disk, the physical address on the disk 24 corresponding to a given logic address stored need not be stored in the mapping segments 34. That is, the position of each logical address in the mapping segment identifies the corresponding physical address. As mentioned above, during initialization of the system 10, the location information stored in mapping segments 34 is read into pointer map 30 prior to the optical disk 24 being available for writing or reading by the host data system user.

To more clearly understand the manner in which location information is stored in the mapping segments 34, please refer to Table I. Table I is an example of the writing of host data and logical address information to a disk 24 that is being used for the first time. It is assumed in this example that a first mapping segment 34, e.g., segment 10, is being filled, that sectors 10-99 are reserved for mapping segments 34, and that host data may be stored in the balance of the sectors on the disk, e.g., sectors 100-100,000. Lastly, assume in this example that the logical memory being emulated by system 10 will provide read/write access to host system 12 of logical addresses 0-999. Consequently, it is conceivable that with 100,000 physical sectors available for 1,000 sectors of logically addressable area, each logical address could be rewritten 100 times by the host system before filling up the optical disk, assuming each logical address is written the same number of times. As also shown, one or more sectors may be reserved for a flaw map, comprising one bit of data for each physical sector to be mapped on the disk. In this map, a zero indicates that the sector is writeable and a one indicating that the sector is flawed and therefore unwriteable. No address data is needed in the flaw map because it stores flow data for each sequential sector on the disk 24 in the same sequential order.

Table I

Host System Determined				
Disk Physical Segment Addr.	Logical Segment Addr.	Data Stored	Mapping Segment 34	Flaw Map
100	000	Data 1	000	0
101	005	Data 2	005	0
102	006	Data 3	006	0
103	Flaw	-	FFFFFF	1
104	Flaw	-	FFFFFF	1
105	500	Data 4	500	0
106	501	Data 5	501	0
107	006	Data 6	006	0
108	000	Data 7	000	0
109	-	-	.	.

As seen in Table I, the first physical segment into which host system user data can be written is, in this example, physical segment 100. If, as its first write operation, the host wishes to write host data identified as "data 1" to logical address 000, the data storage segment at physical location 100 on disk 24 has data 1 written to it along with the logical segment address 000. The first logical address storage portion of mapping segment 34 is also written with the logical address 000.

Note that the number of bytes in mapping segment 34 needed for storage of a logical address depends on the number of locations logically addressable by the specific system. In the preferred embodiment, three bytes of storage space are allocated for each logical address entry in mapping segment 34.

Referring again to Table I, now let us assume that the host user wishes to next store host data comprising data 2 at logical segment address 005. This host data is stored in the next sequential data storage segment address 101 on disk 24 and mapping segment 34 is updated with this new logical address 005. Where the flaw map indicates that a given physical segment is flawed, the controller 14, via flaw map memory 27, described in greater detail below, will skip the flawed segments. Thus, as seen in Table I, the

data storage segments at physical address 103 and 104 are skipped by controller 14 since flaw map 27 indicates that these segments are flawed. An indication that these segments 36 were flawed and skipped is added to the mapping segment 34 map.

An example of how the system 10 takes care of when the host data system 12 user wishes to update data at a particular logical address is also illustrated in Table I. After physical segments at addresses 105 and 106 have been written, host data identified as "data 6" is written to the data storage segment 34 at physical address 197. Its logical address is logical address 006, and mapping segment 34 is updated with this logical address 006. Consequently, if subsequent to this writing transaction, the host data system 12 wishes to read the contents of logical address 006, the disk 24 will read the data storage segment 36 at physical address 107 and not the data at physical address 102.

Table II

RAM Map 30 Actual Addr.	Logical Addr.	Physical Address
X	000	108
X + 1	001	0
X + 2	002	0
X + 3	003	0
X + 4	004	0
X + 5	005	101
X + 6	006	107
X + 7	007	0
.	.	.
.	.	.
.	.	.
X + 500	500	105
X + 501	501	106
.	.	.
.	.	.
.	.	.
X + 999	999	0

To prevent having to read the mapping segments 34 on the disk 24 whenever a read operation is desired, a pointer map 30 is maintained by controller 14 in random access memory. An exemplary pointer map 30 is illustrated in Table II. This table duplicates the data in mapping segment 34 shown in Table I, except that preferably pointer map 30 is oriented by logical address rather than by disk physical segment address. Thus, Table I indicates that logical address 000 was last written with data 7 at physical segment address 108. This physical address 108 is therefore shown in the pointer map 30 illustrated in Table II as being the location to look to for the most recent host data written to logical address 000. Since no data has been written to logical addresses 001-004, no physical address is included in pointer map 30 for these logical addresses. A zero in the physical address column indicates that no data has been written to these logical addresses as yet. Logical address 005 is shown to include data at physical address 101, the last time that this logical address was written to by the host user. Finally, logical address 006 has been written to twice, and the most recent data for this logical address is indicated as being located in the data storage segment 36 at physical address 107. As shown in Table I, data identified as data 6 is stored at this physical address 107.

Also illustrated in Table II, the actual RAM addresses where the logical addresses are maintained can begin at any arbitrary location X. Preferably, the logical addresses are stored in chronological order in the pointer map 30, but this is not required.

Also retained in a random access memory, as mentioned above, is the flaw map memory 27. On initialization of a given disk 24, the segments on the disk 24 containing the flaw map are read by controller 14 and transferred to the flaw map memory 27 to enable its rapid access by controller 14 during a given write transaction. Before any given write to disk operation, or a series of such operations, is performed by the system 10, it is preferred that the controller 14 certify that no new data storage segments 36 have become flawed. To do this, controller 14 causes the optical head 18 to scan the remaining unwritten data storage segments 36 on the disk 24 to determine if any additional defects now exist in any such segment.

An unwritten segment should read as if all zero's were written to the segment. Note that this need to certify that no new flaws have been created on a disk may only need to be performed every month or so, where the system has remained on and the disk has not been removed from the system.

Defect detecting means 26 detects any new defective segment 36 and updates the flaw map memory 5 27 with this information. Flaw map 127 comprises a one bit storage location for each sector or segment 36 addressable on the disk 24. It is organized in the same manner as the flaw map stored on disk 24. Note that the location of flawed segments is never mapped in pointer map 30. Pointer map 30 only points to segments at physical addresses where data has been properly written.

Mapping segment buffer 32 is system 10 stores only the most recent write transactions that have 10 occurred since the contents of buffer 32 were last written onto the disk 24. Buffer 32 is valuable to insure that a given mapping segment 34 is not written to until sufficient transaction data exists to fill the entire segment 34. For example, if a given mapping segment 34 is 512 bytes long, and a given logical address can be stored in 3 bytes, approximately 170 such logical addresses representing 170 writing transactions onto disk 24 can be stored in buffer 32 prior to writing of the contents of buffer 32 to the next mapping 15 segment 34. The advantage of buffer 32 is that it prevents the necessity, after each write transaction, of requiring that the optical head 18 seek the next available mapping segment 34 for storage of the logical address history data relating to that write transaction. This also ensures that, upon initialization, a large number of seeking steps are not required by the controller 14 to find and recover the contents of scattered mapping segments 34 for initializing pointer map memory 30.

Should the mapping segment buffer 32 not be filled when power is lost or when the user removes the 20 disk medium 24 from the system 10, the unwritten information is not lost, nor is the host data lost. Each host data block stored in a respective data storage segment 36 includes its own logical addresses. Each of these logical addresses written subsequent to the last time that write transaction history was taken from buffer 32 and stored on a mapping segment 34, is readable by the controller 14 when the disk 24 is 25 reinserted into the system 10. At that point, the mapping segment buffer 32 after reinitialization of the data from disk 24, will correspond to the buffer 32, prior to power loss or removal of the medium from system 10.

Even where write transaction history has been written to a mapping segment 34, and that segment or series of segments is lost, the system 10 can resurrect this data. In a preferred embodiment of the 30 invention, each mapping segment 34 has its own address which can be read by a controller 14. This address is incremented sequentially for each sequential mapping segment 34 written by the system 10 onto disk 24. Thus, if the mapping segment 34 at sector locations 15 and 16 are lost, for example, upon reinitialization of the disk 24, mapping segment 16 will be found after mapping segment 13. Controller 14 will thus note the loss of the two intervening segments, and will proceed with the resurrection of the 35 contents of these last segments in the same manner as buffer 32 is initialized. The resurrected segment 34 data is recorded at the next sequential unwritten mapping segments 34 on the disk 24. Thus, for example, if 20 mapping segments 34 had been written and mapping segments 14 and 15 had been lost, after this regeneration procedure has occurred, the series of mapping segments 34 would have the following addresses, 13, 16, 17, 18, 19, 20, 14, 15,... Thus, the next time that the mapping segments 34 are read 40 from the disk 24 during a reinitialization, the missing segments 34 will be located later in the mapping segments 34.

As is seen, whenever the system 10 is powered up or a new disk 24 is inserted into the system 10, a initialization routine is performed by controller 14. A flow chart of an exemplary initialization routine is shown in Figure 3.

Referring now to Figure 3, during initialization, controller 14 first reads a given mapping segment 34 45 and fills pointer map 30 with the logical address pointers stored at the first mapping segment 34. The subsequent mapping segments 34 are then read and pointer 30 updated with the logical address information stored in each of these segments 34. Note that the controller 14 knows that the first entry in the first mapping segment 34 corresponds to the first physical address location of data storage segments 36 on 50 that particular disk 24.

Once all of the mapping segments 34 have been read, the last physical segment 36 that contains host data is determined whose logical address is stored in a mapping segment 34. The controller 14 then searches for subsequent sequential data storage segments that have been written to with data by the host 55 data system 12 whose logical addresses are not included in a mapping segment 34. Where data is found, the logical address of that data is added to both pointer map 30 and buffer 32. The controller 14 then increments by one to look at the next sequential physical segment to again test to see if any data is stored in that segment 36. Once all of the physical segments have been sequentially tested in this manner and the next data storage segment is indicated as not having any data therein, this indicates that the initialization

process has been completed by controller 14. The result of this process is the rebuilding the entire sector buffer 32 and complete updating of pointer map 30 for enabling accurate access to all most recent host data updates to the logical addresses maintained on disk 24.

After the initialization of the system, controller 14 knows the address of the next unwritten data storage segment 36 available for use for storing the next host data brought from the host data system 12. Note also that controller 14 keeps track of the total number of physical sectors available for writing so that when all remaining sectors have been written or contain flaws, controller 14 is able to indicate disk full status.

Referring now to Figure 4, shown therein is a flow chart of the process for writing data to the disk 24. As seen in Figure 4, controller 14 first locates the next sequential free data storage segment on disk 24 that is not flawed. The host data is then written to this data storage segment 36 along with the logical address of the host data. The write transaction is then written to the mapping segment buffer 32 to update the write transaction history stored therein for later recording on the disk 24 at the next mapping segment 34. The logical address is also used to update the pointer map memory 30. Controller 14 then looks to see if buffer 32 is full. If buffer 32 is not full, this completes the writing operation. If the buffer is full, the next unwritten mapping segment 34 is located and the contents of buffer 32 is flushed to this next mapping segment. Once this operation is completed, the writing operation ends.

Figure 5 illustrates the operation of controller 14 when a read command is received by the system 10. As seen in Figure 5, the host data system 12 couples to the system 10 a logical address that it wishes to receive host data from. This is received by controller 14 and is used to extract the data storage segment 36 physical address from the pointer map memory 30 corresponding to this logical address for the most recently written host data thereto. If the entry in pointer map memory 30 is a zero, a null entry, the controller 14 will transmit blank data to the host system 12. If the entry is not zero, the optical head 18 is caused to seek to the physical address of the data storage segment 36 which was found in the above steps to obtain the most recently written host data for the specified logical address. The controller 14 then reads the data stored at the specified segment 36 along with the logical address stored in the selected data storage segment. Then, if the detected logical address are not the same as that requested by the host system 12, the controller 14 retries to read the specified physical sector. If the logical addresses match, the host data at the designated data storage segment is read and transmitted back to the host data system 12.

As mentioned above, alternative ways of storing write transaction history in mapping segment 34 and in pointer map 30 are contemplated. For example, the host data system may elect at the beginning of writing of a particular disk to format all write transactions so as to include four data storage segments or some other number of data storage segments for each write operation. As a result, in a disk 24 having identical numbers of sectors, if only one fourth of the number of normal data storage segments need be kept track of, the corresponding size of the pointer map 30 is also reduced by four. This is also important, for example, in a conventional optical disk 24 having over 700,000 sectors, since it would reduce the number of mapping segments 34 needed to maintain a history of the writing of data to these sectors.

Where a series of flawed sectors are grouped in sequence, e.g., the sectors at physical locations, 30-33 are flawed, the representative of these flawed sectors may also be reduced by storing a record of such grouped flawed sectors in a single 3 byte entry in the mapping segments 34. As part of this entry, 1 byte of data would indicate the number of flawed sectors in the group.

The system described provides an improved optical data storage system for emulating a read/write magnetic disk. The system and the method of its use provide means to detect and bypass defects in the recording surface of the optical disk, and to permanently retain data without erasure. The system further provides for rapid identification and retrieval of stored data stored at any one of a plurality of logical addresses as well as for protection of disk's data directory from loss.

Claims

1. An optical data storage system (10), for enabling the writing of data to and the reading of data from a write-once, read-many optical disk (24) by a host system (12) such that said host system (12) is enabled to access any one of a predetermined number of logical addresses for storing host data whether or not host data has previously been stored at that logical address, characterised by a write-once optical disk (24) including at least one recording surface (22) divided into a plurality of sequential data storage segments (36), each having a host data portion and a logical address portion, and having the physical location of each of said data storage segments (36) represented by a physical address different from the physical address of any other data storage segment (36); and means (16, 14, 18) for writing each successive host data to the

host data portion of a next sequential unwritten data storage segment (36), including means for writing to the logical address portion of said next sequential data storage segment the logical address specified by said host system (12) for said host data.

2. A system as claimed in Claim 1, characterised in that said optical disk (24) includes a plurality of sequential mapping segments (34) each for storing at least one logical address; and means responsive to said host data writing means for writing to the next sequential unwritten mapping segment (34) the logical address specified by said host data system (12) for said host data being written by said host data writing means.

3. A system as claimed in Claim 1 or Claim 2, characterised by means for retrieving host data stored at a logical address specified by said host system (12), including means for determining the last physical address into which host data was written having said specified logical address, and means for reading the host data stored in the data storage segment (36) at said last physical address.

4. A system as claimed in Claim 3, characterised in that said means for determining the last physical address into which host data was written having a specified logical address comprises pointer map memory means (30) for storing a map of all available host data logical addresses and their correspondence to the most recent physical addresses of the data storage segments (36) of said optical disk (24) wherein host data has been written, said means for writing host data including means for updating said pointer map memory means (30) with the physical address of said next sequential data storage segment (36) such that the logical address specified for said host data is associated in said pointer map memory means (30) with the physical address of said next sequential data storage segment (36).

5. A system as claimed in Claim 4, characterised by means for initializing said pointer map memory means (30), said initializing means including means for reading said mapping segments (34) and for storing in said pointer map memory means (30) for each logical address addressable by said host system (12) the physical address of the most recent data storage segment (36) wherein host data to be stored at said logical address has been stored on said disk (24), said initialization means being operative upon power-up of said system (10) and each time a new optical disk (24) is inserted into said system.

6. A system as claimed in Claim 5, characterised in that initializing means comprises means for generating the physical address of a data storage segment (36) for each logical address stored in said mapping segments (34).

7. A system as claimed in anyone of Claims 4 to 6, characterised in that said pointer map memory means (30) comprises a random access memory.

8. A system as claimed in any preceding claim, characterised by means (27) for detecting and storing the location of any flawed data storage segments (36) in said recording surface (22) of said optical disk (24) and means responsive to said defect storing means (27) for preventing host data from being written to any said flawed data storage segment (36).

9. A system as claimed in Claim 8, characterised by means for updating said flaw data storing means (27) with the location of any new flaws detected in said recording surface (22) of said optical disk (24).

10. A system as claimed in Claim 2, or any one of Claims 3 to 9 as dependent on Claim 2, characterised by means responsive to said host data writing means for temporarily storing the logical address specified by said host system (12) for said host data being written by said host data writing means; and means for transferring the contents of said logical address storing means to the next sequential unwritten mapping segment (34) on said disk (24) after a predetermined number of logical addresses have been stored in said storing means.

FIG. 1

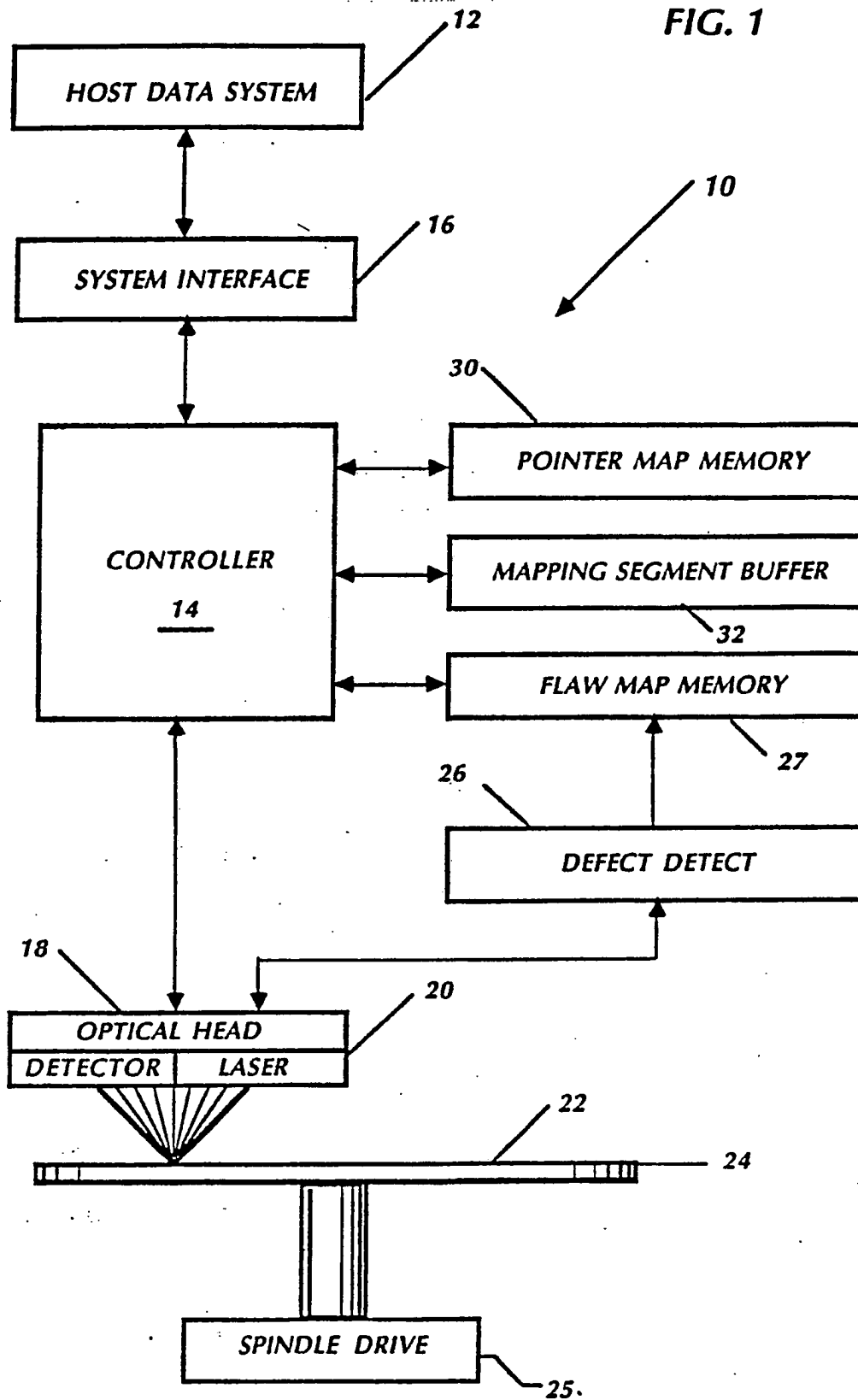


FIG. 2

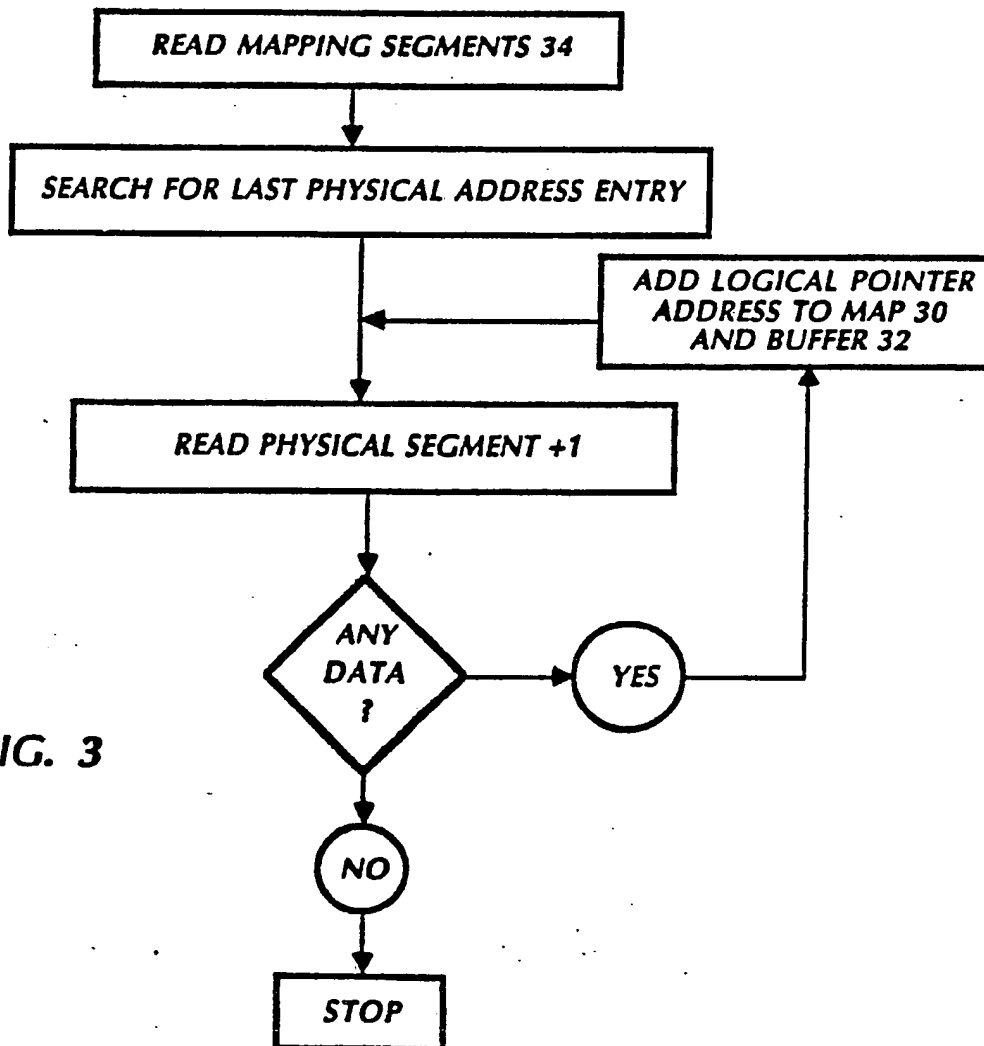
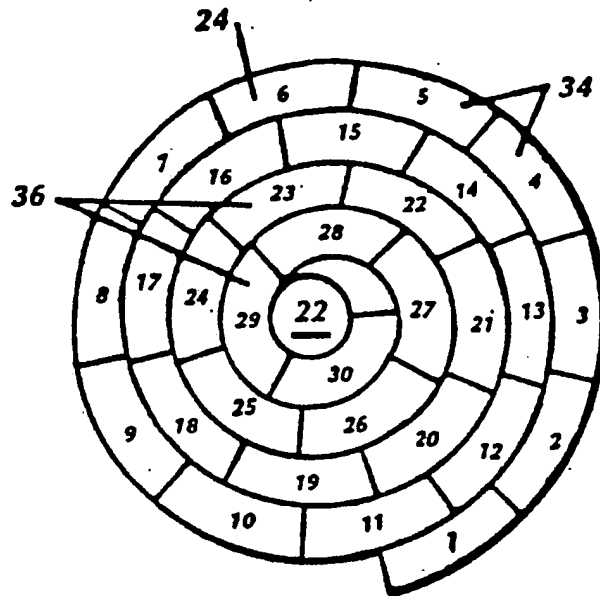
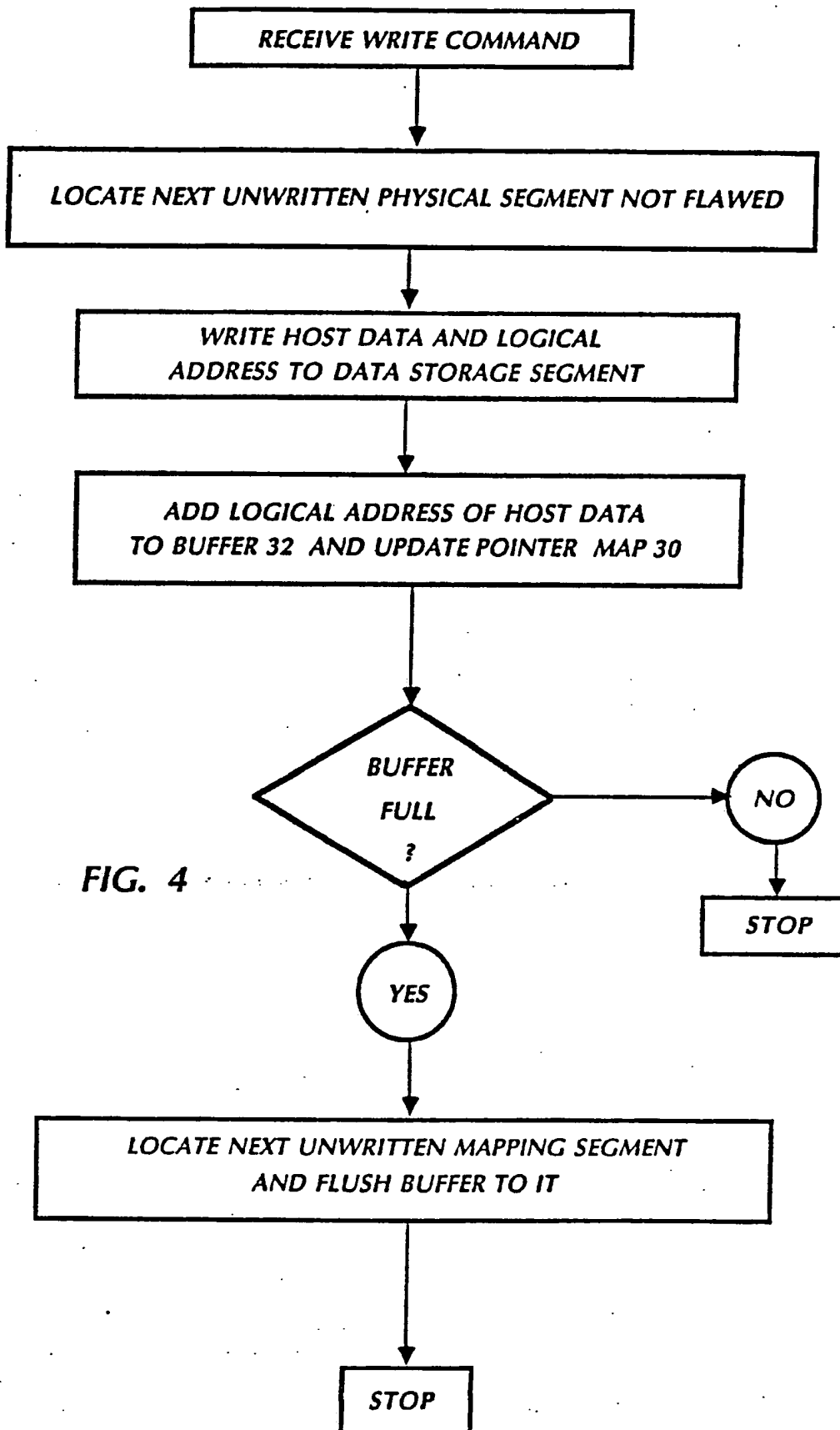
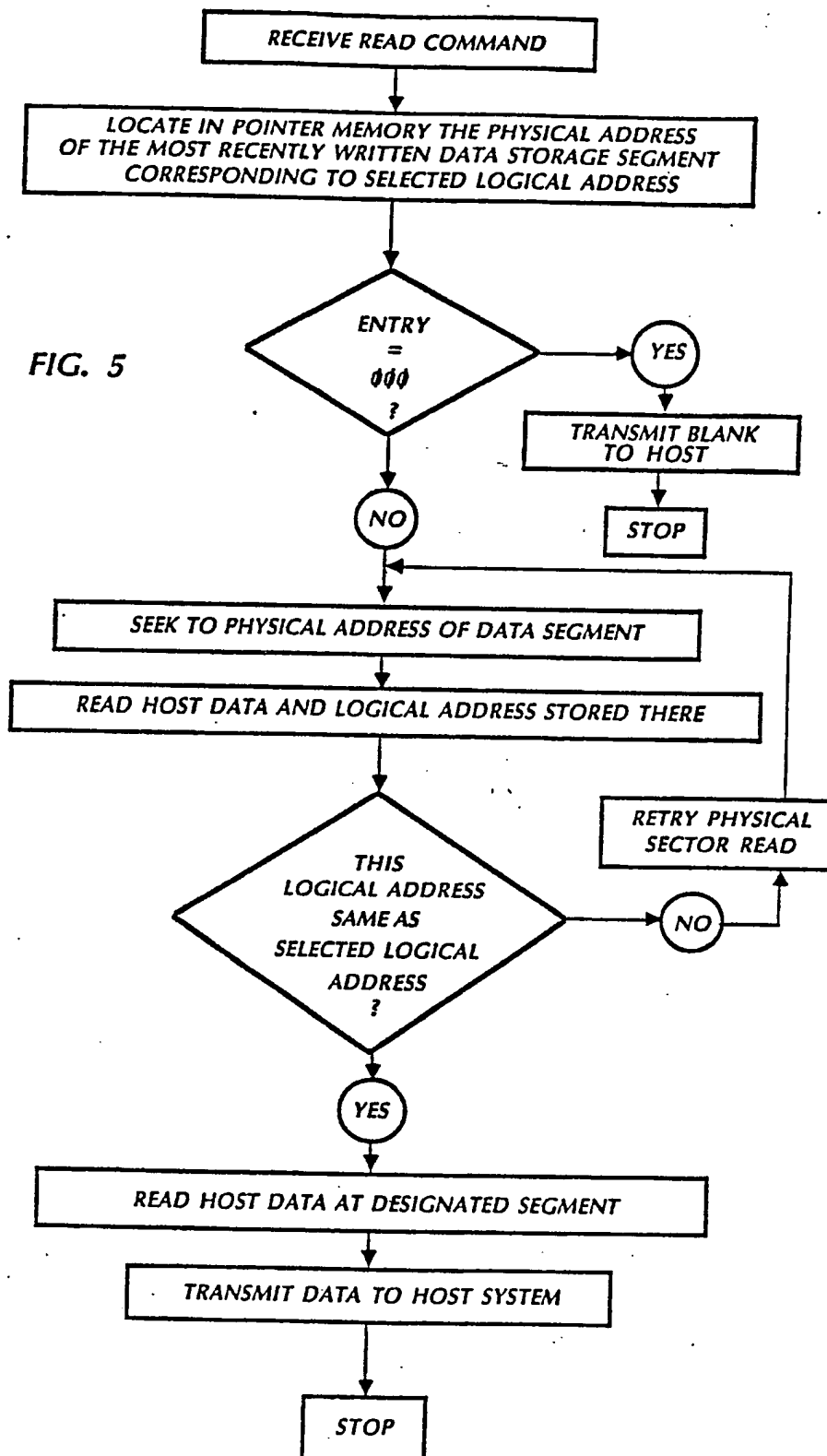


FIG. 3







European Patent
Office

EUROPEAN SEARCH REPORT

Application Number

EP 88 30 0645

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.4)
X	US-A-4 633 393 (RUNDELL) * Column 2, line 35 - column 3, line 15; column 4, lines 37-59; column 4, line 65 - column 5, line 49 *	1,3	G 11 B 7/013 G 11 B 20/12 G 11 B 27/28
Y	* Whole document *	2,8-10	
A	---	4-7	
X	US-A-4 633 391 (RUNDELL) * Column 4, lines 37-59; column 4, line 65 - column 5, line 49 *	1,3	
Y	* Whole document *	2,8-10	
A	---	4-7	
Y	PATENT ABSTRACTS OF JAPAN, vol. 11, no. 5 (P-533)[2452], 8th January 1987; & JP-A-61 182 674 (RICOH CO., LTD) 15-08-1986 * Abstract *	2,10	
A	IDEM	1	
Y	PATENT ABSTRACTS OF JAPAN, vol. 10, no. 341 (P-517)[2397], 18th November 1986; & JP-A-61 142 571 (RICOH CO., LTD) 30-06-1986 * Abstract *	2,10	TECHNICAL FIELDS SEARCHED (Int. Cl.4) G 11 B G 06 F
A	IDEM	1	
	---	-/-	
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 21-09-1988	Examiner DAALMANS F.J.
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons ----- & : member of the same patent family, corresponding document			



DOCUMENTS CONSIDERED TO BE RELEVANT				
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.4)	
Y	PATENT ABSTRACTS, vol. 12, no. 224 (P-721)[3071], 25th June 1988; & JP-A-63 18 524 (RICOH CO., LTD) 26-01-1988 * Abstract *	8,9		
A	IDEM ---	1		
Y	PATENT ABSTRACTS OF JAPAN, vol. 11, no. 287 (P-617)[2734], 17th September 1987; & JP-A-62 84 433 (HITACHI ELECTRONICS ENG. CO., LTD) 17-04-1987 * Abstract *	8,9		
A	IDEM ---	1		
A	OPTICA' 87, THE INTERNATIONAL MEETING FOR OPTICAL PUBLISHING AND STORAGE, Amsterdam, 14th-16th April 1987, pages 49-58, Learned Information, Oxford, GB; F. DIRSCHIEDL et al.: "File system for write-once optical disks" * Whole article *	1-6,10		TECHNICAL FIELDS SEARCHED (Int. Cl.4)
A	OPTICA' 87; THE INTERNATIONAL MEETING FOR OPTICAL PUBLISHING AND STORAGE, Amsterdam, 14th-16th April 1987, pages 331-338, Learned Information, Oxford, GB; G. RUSSO et al.: "Optical storage for archiving astronomical data" * Whole article *	1-4,7		
A	EP-A-0 165 382 (IBM CORP.) * Whole document * --- -/-	1-3		
The present search report has been drawn up for all claims				
Place of search THE HAGUE		Date of completion of the search 21-09-1988	Examiner DAALMANS F.J.	
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons</p> <p>& : member of the same patent family, corresponding document</p>				



DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl. 4)
A	EP-A-0 072 704 (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD) * Whole document *	1,3	
A	SPIE, ADVANCES IN LASER SCANNING TECHNOLOGY, vol. 299, 1981, pages 33-39, Bellingham, Washington, US; S.L. CORSOVER et al.: "Error management techniques for optical disk systems" * Whole article *	1,3,8	
A	EP-A-0 227 530 (PICARD) * Whole document *	1,3	
A	PATENT ABSTRACTS OF JAPAN, vol. 9, no. 214 (P-384)[1937], 31st August 1985; & JP-A-60 74 158 (MATSUSHITA DENKI SANGYO K.K.) 26-04-1985 * Abstract *	1,3	
A	PATENT ABSTRACTS OF JAPAN, vol. 11, no. 335 (P-632)[2782], 4th November 1987; & JP-A-62 117 180 (CANON INC.) 28-05-1987 * Abstract *	1,3-6	
A	PATENT ABSTRACTS OF JAPAN, vol. 10, no. 74 (P-439)[2131], 25th March 1986; & JP-A-60 212 886 (NIPPON DENKI K.K.) 25-10-1985 * Abstract *	1,3-6	
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 21-09-1988	Examiner DAALMANS F.J.
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons</p> <p>& : member of the same patent family, corresponding document</p>			